Heuristic Rule-Based Regression via Dynamic Reduction to Classification

TECHNISCHE UNIVERSITÄT DARMSTADT

Frederik Janssen and Johannes Fürnkranz
Technical University Darmstadt
{janssen,juffi}@ke.tu-darmstadt.de

1 Introduction

The setting in this work is

- regression datasets, i.e., prediction of numerical target variable
- simple **IF-THEN** rules should be learned that predict a single value, and can be used as decision list
- two approaches to learn regression rules:
  - either transform regression dataset to classification dataset, or
  - **directly learn rules on regression dataset** ← considered here
- rules are learned by a simple separate-and-conquer algorithm [1]

2 Rule Learning Heuristics

Rule Learning Heuristics are the most important part of a separate-and-conquer algorithm. In this work, we used the following heuristics:

- laplace (lap) $= \frac{p+1}{p+n+2}$ — known to overfit
- weighted relative accuracy (wra) $= \frac{p}{P} - \frac{n}{N}$ — known to underfit
- correlation (corr) $= \frac{p \cdot N - n \cdot P}{\sqrt{P \cdot N \cdot (p+n) \cdot (P-p+N-n)}}$ — stable heuristic (cf. [2])
- relative cost (rcm) $= c \cdot \frac{p}{P} - (1-c) \cdot \frac{n}{N}$ — with parameter $c = 0.342$ as suggested in [2]

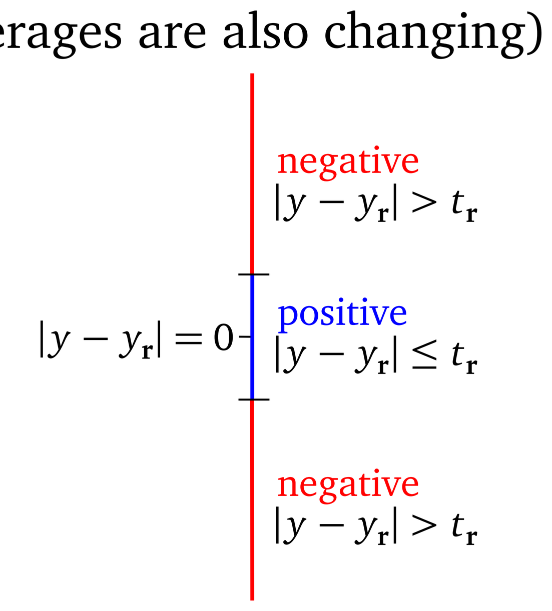
3 Dynamic Reduction to Classification

- In regression datasets, there is no notion of positive and negative examples (as they only have numbers as target variable)
- **idea:** label all examples that are within the standard deviation (σ) of the rule's prediction as positive and all that are outside as negatives
- implemented with a threshold $t_\mathbf{r}$ = factor $\cdot\, \sigma_\mathbf{r}$ (subscript $\mathbf{r}$ added as these values may change for each refinement as the coverages are also changing)

$$t_\mathbf{r} = \text{factor} \cdot \sigma_\mathbf{r}$$

$$class(x) = \begin{cases} positive & \text{if } |y - y_\mathbf{r}| \leq t_\mathbf{r} \\ negative & \text{if } |y - y_\mathbf{r}| > t_\mathbf{r} \end{cases}$$

where x is the current example, y is the true value of the example x, and $y_\mathbf{r}$ is the value predicted by rule $\mathbf{r}$.



- there are different ways of defining the threshold $t_\mathbf{r}$, but as mentioned above we experimented with the standard deviation, and also tried to slightly increase or decrease it (by setting factor = 0.95 and factor = 1.05)
- the total number of positive and negative examples is defined as

$$P_\mathbf{r} = \sum_{i=1}^{m} \mathbf{1}(|y_i - y_\mathbf{r}| \leq t_\mathbf{r}), \qquad N_\mathbf{r} = m - P_\mathbf{r}$$

where m = number of examples, and $\mathbf{1}(.)$ is the indicator function.

- Stopping Criterion: stop learning when 90% of the examples are covered

4 Algorithm Setup

We compared Dynamic Reduction to Classification with a variety of other regression algorithms:

- Other Rule-based regression algorithms
  - M5RULES [3] in default mode and with prediction of single value (-R)
  - REGENDER [4], in default configuration (50 rules), and in setting recommended by the authors (200 rules, different loss function, and different optimization technique)
- Other standard regression algorithms
  - LINEAR REGRESSION, MULTILAYER PERCEPTRON, and SVMREG
- Static reduction to classification
  1. discretize the class variable (equal-frequency)
  2. use a classification-version of our rule learner on the discretized data

We also evaluated bagged versions of our algorithm in order to reduce its restriction to piecewise constant predictions.

5 Datasets

We used 16 regression datasets from the UCI Repository and Luis Torgos webpage (http://www.liaad.up.pt/~ltorgo/Regression/DataSets.html). The focus was to select datasets that have a lot of disjunct target values.

6 Results

Dynamic Regression by Classification					
factor	heuristic	*rrmse*	rank	# rules	# conds
0.95	wra	0.752	8.63	15.06	38.31
0.95	lap	0.784	11.19	11.25	13.88
0.95	corr	0.726	6.50	10.13	24.63
0.95	rcm	0.780	9.81	19.06	34.25
1.00	wra	0.764	10.06	17.06	47.81
1.00	lap	0.774	10.63	10.19	12.50
1.00	corr	0.753	8.38	9.25	22.06
1.00	rcm	0.767	9.50	19.06	35.75
1.05	wra	0.780	13.13	13.19	34.19
1.05	lap	0.772	10.19	9.69	11.81
1.05	corr	0.796	12.88	10.25	33.31
1.05	rcm	0.775	9.75	19.44	37.56
Static Regression by Classification					
# classes	heuristic	*rrmse*	rank	# rules	# conds
5	wra	0.883	18.25	5.63	20.75
5	lap	0.857	14.75	84.56	197.44
5	corr	0.844	15.13	28.06	84.00
5	rcm	0.852	16.63	22.88	68.00
10	wra	0.930	18.69	6.06	23.13
10	lap	0.872	17.00	138.44	339.25
10	corr	0.864	15.88	49.31	167.25
10	rcm	0.901	17.94	20.75	67.31
20	wra	0.965	20.81	10.06	36.56
20	lap	0.872	18.06	177.44	423.63
20	corr	0.862	17.81	95.13	295.00
20	rcm	0.928	19.13	33.19	102.13
Other Rule-Based Regression algorithms					
algorithm		*rrmse*	rank	# rules	# conds
REGENDER (50)		0.768	9.38	50.00	190.00
M5RULES -R		0.773	10.44	6.19	14.94

Table 1: Evaluation of dynamic regression by classification (top), static regression by classification (bottom), and two other rule-based learning algorithms.

algorithm	*rrmse*	rank	# rules	# conds
Regular	0.726	7.06	10.13	24.63
Bagged (10)	0.671	5.88	97.94	245.81
Bagged (20)	0.659	4.94	186.75	451.25
Bagged (50)	0.658	4.63	465.88	1146.6
LR	0.651	4.31	—	—
MLP	0.746	5.88	—	—
SVMreg	0.673	5.19	—	—
RegENDER	0.679	4.50	200.00	1163.6
M5Rules	0.604	2.63	2.94	5.38

Table 2: Comparison of a bagged version to other types of regression algorithms

- *correlation* with a factor of 0.95 is the best choice among the configurations
- the dynamic approach is able to outperform the static one significantly (best setting outperforms all but two static approaches with $p = 0.1$)
- preferences of the heuristics known from classification do not carry over to the dynamic approach (i.e., *laplace* finds fewer rules than *wra*)
- bagged versions of the algorithm work comparable to state-of-the-art algorithms (cf. Table 2 and Figure 1)

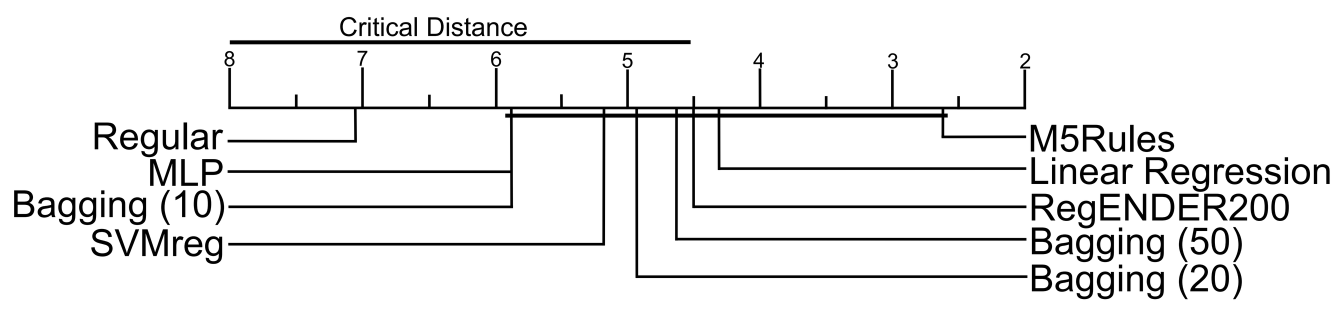


Figure 1: Comparison of the algorithms shown in Table 2 against each other with the Nemenyi test. Groups of algorithms that are not significantly different (at $p = 0.01$) are connected.

7 Conclusion and Future Work

- Dynamic Reduction to Classification allows to use classification heuristics directly
- Dynamic Reduction to Classification outperforms the Static Approach (a priori discretization of class variable)
- Dynamic Approach is en par with other rule-based regression algorithms

References

[1] J. Fürnkranz. Separate-and-conquer rule learning. *Artificial Intelligence Review*, 13(1):3–54, February 1999.

[2] F. Janssen and J. Fürnkranz. On the quest for optimal rule learning heuristics. *Machine Learning*, 78(3):343–379, March 2010.

[3] G. Holmes, M. Hall, and E. Frank. Generating rule sets from model trees. In *Proc. 12th Australian Joint Conference on Artificial Intelligence (AI-99)*, pp. 1–12. Springer, 1999.

[4] K. Dembczyński, W. Kotłowski, and R. Słowiński. Solving regression by learning an ensemble of decision rules. In *Proc. 9th International Conference on Artificial Intelligence and Soft Computing (ICAISC-08)*, pp. 533–544, Zakopane, Poland, 2008. Springer-Verlag.